I Linked Open Data: crearli a partire da una fonte open, ben descritta e strutturata

Giorgia Lodi
Tecnologa del Laboratorio di Tecnologie Semantiche (STLab)
dell'Istituto di Scienze e Tecnologie della Cognizione del CNR

Sommario

- Cosa sono i linked (open) data
  - Standard RDF
- Quali strumenti esistono per produrre linked (open) data
  - RML- RDF Mapping Language
- Il caso dei Linked Open Data dei finanziamenti del PNRR

Sommario

- Cosa sono i linked (open) data
  - Standard RDF
- Quali strumenti esistono per produrre linked (open) data
  - RML- RDF Mapping Language
- Il caso dei Linked Open Data dei finanziamenti del PNRR

Il Web

Risorse: Documenti Web

HTML è lo standard usato **per costruire pagine Web. Le pagine** possono essere **collegate attraverso dei link non tipati** (href)

Domanda: se facessimo la stessa cosa sui dati introducendo tipi e quindi semantica?

RDF - Resource Description Framework (1/2)

- Un modello standard per i dati **NEL** web dove i dati sono espressi **sottoforma di triple**
  - `Soggetto Predicato Oggetto`
- Ogni entità è identificata **univocamente da un URI** (Uniform Resource Identifier)
  - URI - una sequenza **univoca di caratteri** che consente di identificare e distinguere una risorsa dall'altra nel Web (identificatore)

RDF

RDF - Resource Description Framework (1/2)

- Il **Soggetto** e il **Predicato** hanno **sempre un URI**
  - Esistono casi di uso di cosiddetti "blank node" (dove un nodo non ha un URI) ma nei dati è meglio evitarlo
- L'**Oggetto** può anche essere un **valore predefinito** (Literale)
- Le **triple sono interconnesse** quando **condividono la stessa entità** (l'oggetto di una tripla è il soggetto di un'altra tripla)
  - Il risultato è un **grafo interconnesso** di triple (Linked Data)

Grafo interconnesso (Knowledge graph)

https://dbpedia.org/resource/Bologna

https://dbpedia.org/resource/Italy

https://dbpedia.org/ontology/country

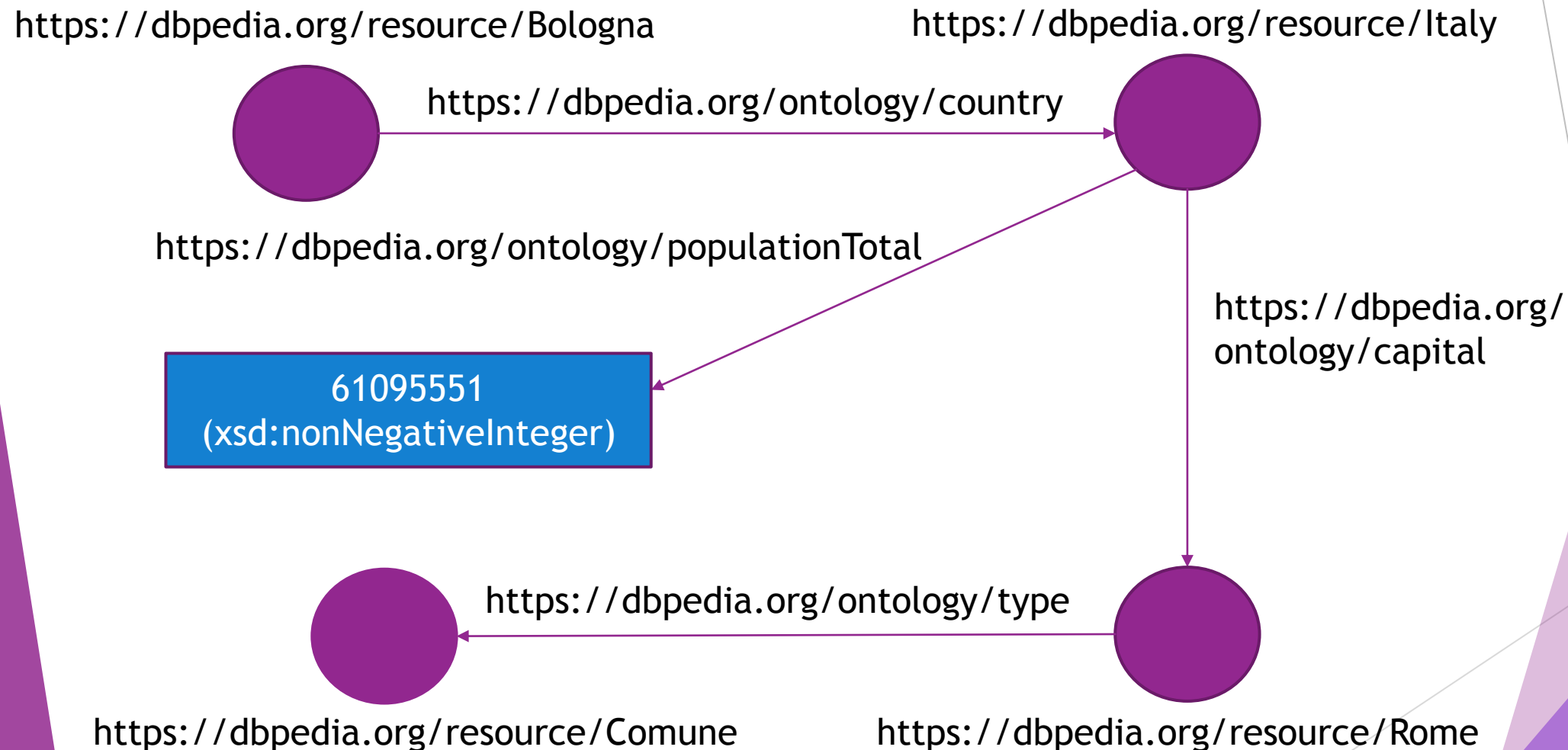
https://dbpedia.org/ontology/populationTotal

https://dbpedia.org/ontology/capital

61095551 (xsd:nonNegativeInteger)

https://dbpedia.org/ontology/type

https://dbpedia.org/resource/Comune

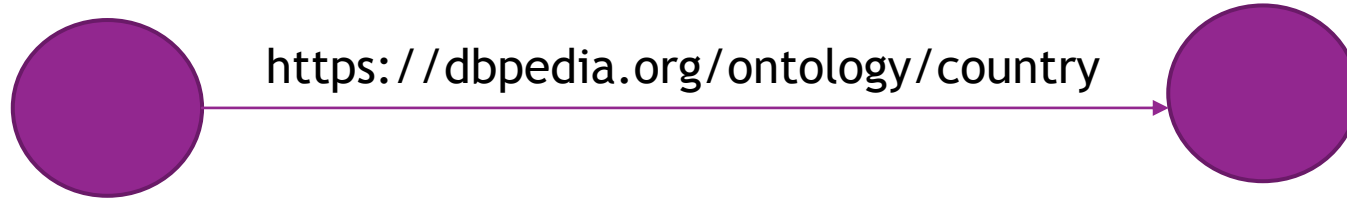
https://dbpedia.org/resource/Rome

Dati tabellari vs Linked (Open) Data

- I **dati tabellari** vi danno una **visione su singole cose** e spesso senza specificare un significato
  - Intestazione di colonna con qualche spiegazione sono il massimo che si ottiene (con qualche eccezione)
- I Linked (Open) Data pongono invece l'accento sulle relazioni tra diverse cose, anche definite potenzialmente da diversi attori in diversi dataset (il dato è una rete)
- I Linked (Open) Data definiscono un significato per le cose (oggetti reali del mondo) e per le relazioni tra le cose
  - Il significato è machine readable = un software può interpretarlo
- I Linked (Open) Data utilizzano standard consolidati/noti del Web (Semantico) → Sono **Interoperable**

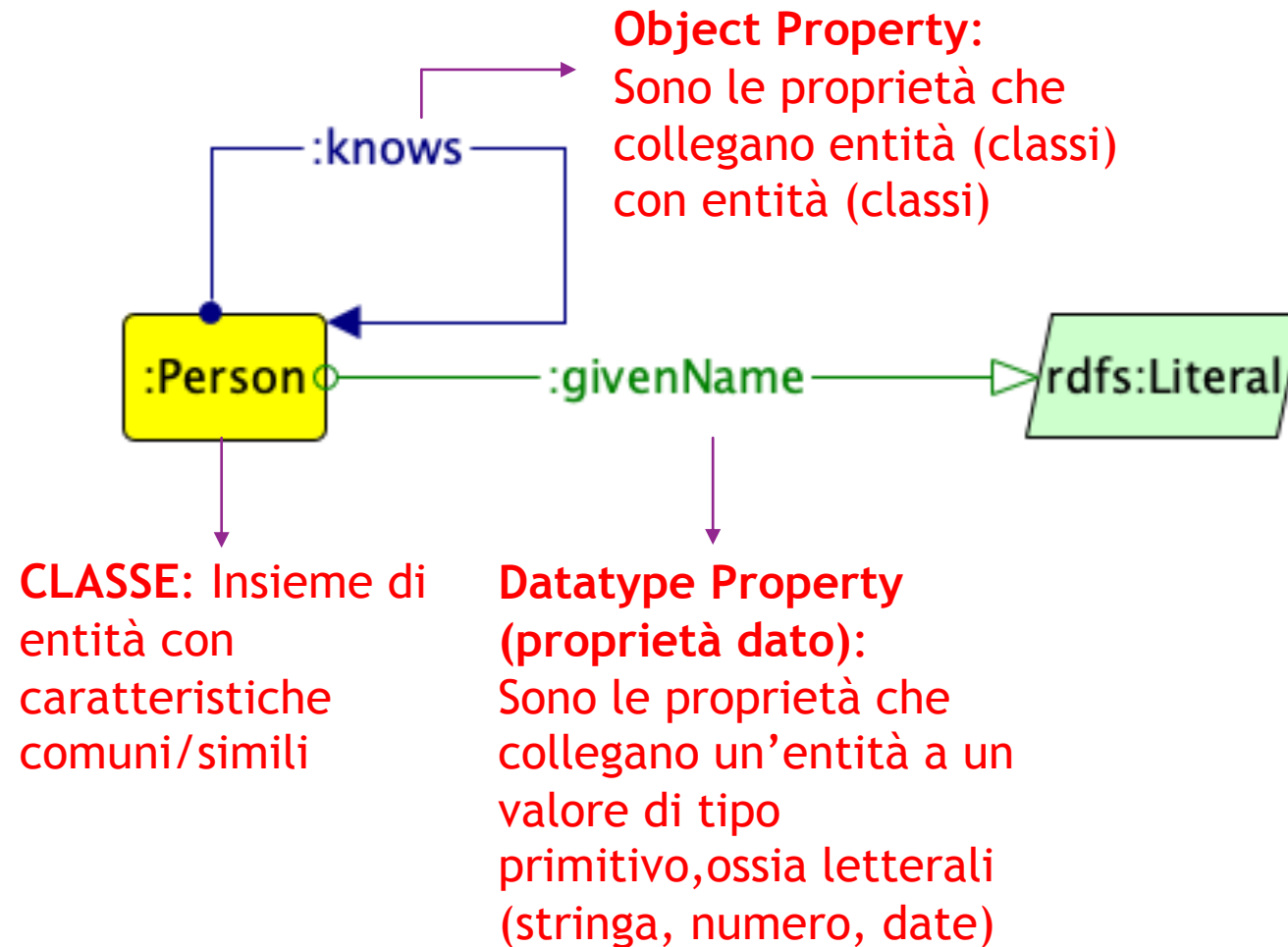
Il significato dei dati

https://dbpedia.org/resource/Bologna

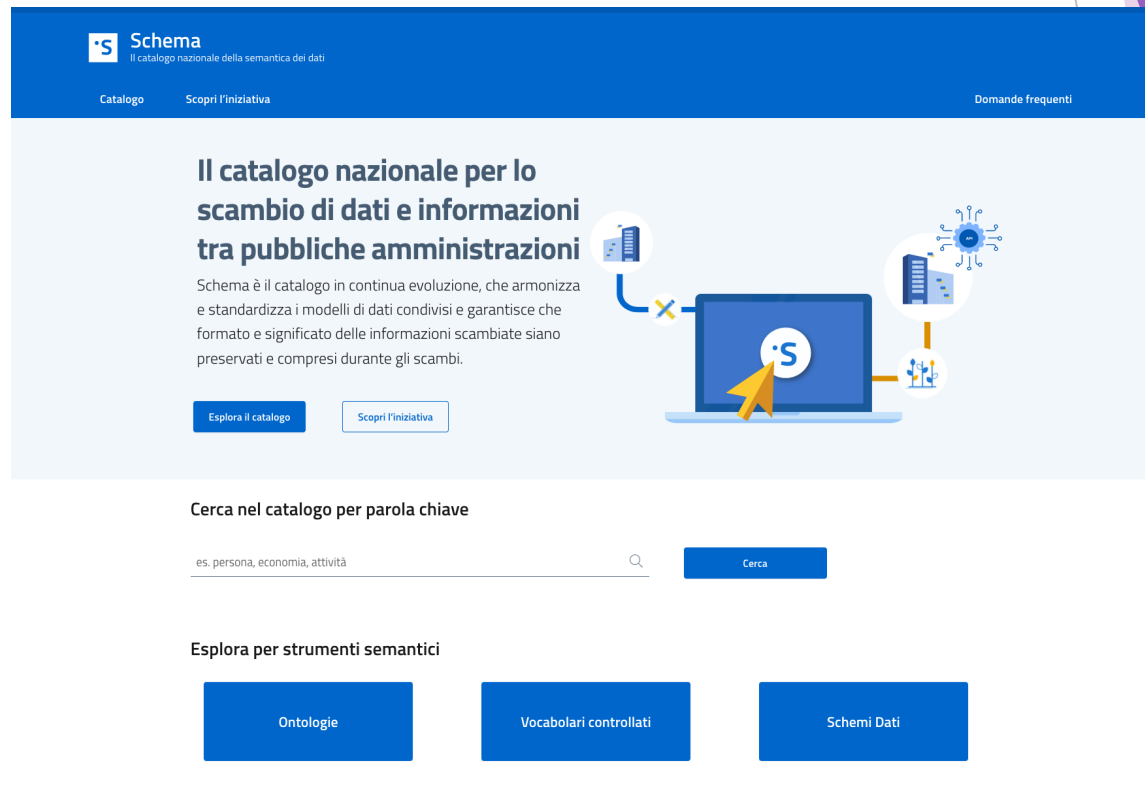
https://dbpedia.org/resource/Italy

https://dbpedia.org/ontology/country

- La domanda da porsi è come rappresentare il significato delle relazioni e dei tipi associati ai nodi delle triple

- Risposta: Si definiscono **ontologie** attraverso l'uso di standard del Web Semantico appositamente nati per questo
  - Le ontologie sono rappresentate anch'esse come grafi di nodi (**classi**) e di archi (**proprietà**/predicati) tra nodi
  - Gli standard che si usano sono **RDFSchema** e **OWL** (Ontology Web Language)

Esempio di ontologia

Dove trovo le ontologie in Italia?

Nel catalogo della semantica nazionale

Realizzato attraverso la piattaforma **schema.gov.it** e utilizzato dalla **Piattaforma Digitale Nazionale Dati (PDND)** per garantire che il formato e il significato dei dati scambiati si preservi durante gli scambi

Serializzazioni di RDF (dei Linked Open Data)

- Ci sono diverse serializzazioni o formati per lo standard RDF. I più utilizzati sono
  - **RDF/Turtle** (file con estensione .ttl)
    - ✓ formato molto compatto a oggetti
    - ✓ leggibile facilmente da persone
  - N3 (file con estensione .n3)
    - ✓ formato molto prolisso dove tutti gli URI sono esplicitati per esteso per soggetto, predicato e oggetto
  - RDF/XML (file con estensione .rdf)
    - ✓ formato simile a XML ma per lo standard RDF
  - JSON-LD (file con estensione .jsonld)
    - ✓ la rappresentazione RDF nel mondo JSON
    - ✓ Specifica due elementi importanti, il contesto e il grafo

Esempio RDF-Turtle

```
@prefix cpv: <https://w3id.org/italia/onto/CPV/> .

<https://w3id.org/italia/data/person/12345>
   a cpv:Person ;
   cpv:givenName "Giorgia" ;
   cpv:familyName "Lodi" ;
   cpv:hasBirthPlace
<https://w3id.org/italia/controlled-
vocabulary/territorial-classifications/cities/037006-
(1937-12-06)> .
```

Il pattern che si è adottato in Italia per definire gli URI, sulla base del documento Europeo 10 Rules for Persistent URIs, è il seguente:

https://{dominio}/{tipo}{di cosa parliamo}/{codice locale}

Sommario

- Cosa sono i linked (open) data
  - Standard RDF
- Quali strumenti esistono per produrre linked (open) data
  - RML- RDF Mapping Language
- Il caso dei Linked Open Data dei finanziamenti del PNRR

Come creare Linked (Open) Data

- Da
  - Testo non strutturato (e.g., Natural Language Processing)
  - Dati tabellari (da file CSV)
  - Da dati scritti come coppie chiave-valore (JSON)
  - Da dati strutturati rappresentati con lo standard XML
  - Da database relazionali (tabelle tra loro collegate)
  - By scratch ☺

Strumenti per crearli da file tabellari esistenti

- Gli strumenti più noti nell'ambito della comunità scientifica di riferimento sono
  - **RML - RDF Mapping Language [1]** - basato sullo standard del W3C R2RML [2] (quest'ultimo consente di trasformare dati in database relazioni a grafi RDF)
    - ✓ Basato sullo standard RDF
    - ✓ Estende R2RML con una propria piccola ontologia per supportare la trasformazione in grafo RDF di una serie di formati
  - **SPARQL Anything [3]**
    - ✓ Non così diffuso come il caso RML ma sta prendendo piede
    - ✓ Basato sullo standard SPARQL, protocollo e linguaggio standard del W3C per l'interrogazione dei Linked Open Data

[1] https://rml.io/specs/rml/

[2] https://www.w3.org/TR/r2rml/

[2] https://sparql-anything.cc/

RML - funzionalità

EXTRACTION

- Definizione delle sorgenti da cui partire per la trasformazione dei dati

- Uno stesso file di regole di mappatura può contenere più sorgenti di dato da cui partire

RML - funzionalità

TRANSFORMATION

- È possibile fare bonifica sui dati originali attraverso l'uso di funzioni che possono maneggiare stringhe, numeri o introdurre elementi di programmazione (if then else)
- Le funzioni sono gestite mediante l'uso di un'ontologia che si chiama FNO (Function Ontology)
  - string_contains
  - normalizeDate (normalizeDateTime)
  - isNull
  - controls_if (se una condizione è vera o falsa)
  - string_md5
  - slugify
  - trueCondition
  - array_join

RML - funzionalità

- È possibile combinare dati che sono in un dataset CSV con dataset anche disponibili in altri formati (e.g., JSON)

- Questo può avvenire usando delle cosiddette **JoinCondition** che consentono di verificare se un certo dato in un dataset fa match con lo stesso dato in un altro dataset

- Supporta due serializzazioni: RDF/Turtle e JSON-LD

Motori di trasformazione RML

RML Mapper [1]

- Interamente sviluppato in Java da coloro che hanno sviluppato la specifica RML (University of Gent)
- Open Source
- Pieno di test case con svariati esempi anche sull'uso delle funzioni

PyRML [2]

- Interamente sviluppato in Python con alla base PANDAS
- Open Source (prodotto dal laboratorio STLab dell'ISTC-CNR)
- Recenti sviluppi consentono di supportare anche le funzioni come definite in FNO

[1] https://github.com/RMLio/rmlmapper-java

[2] https://github.com/anuzzolese/pyrml

Alcuni suggerimenti

- Quando i file da convertire sono complessi e coinvolgono anche diversi concetti è utile predisporre regole di mapping per piccole porzioni del dato
  - Facilità di debugging
  - Facilità di manutenzione delle varie risorse digitali

- Più file di regole di mappatura possono essere usati per produrre un unico file secondo una qualche serializzazione RDF

Sommario

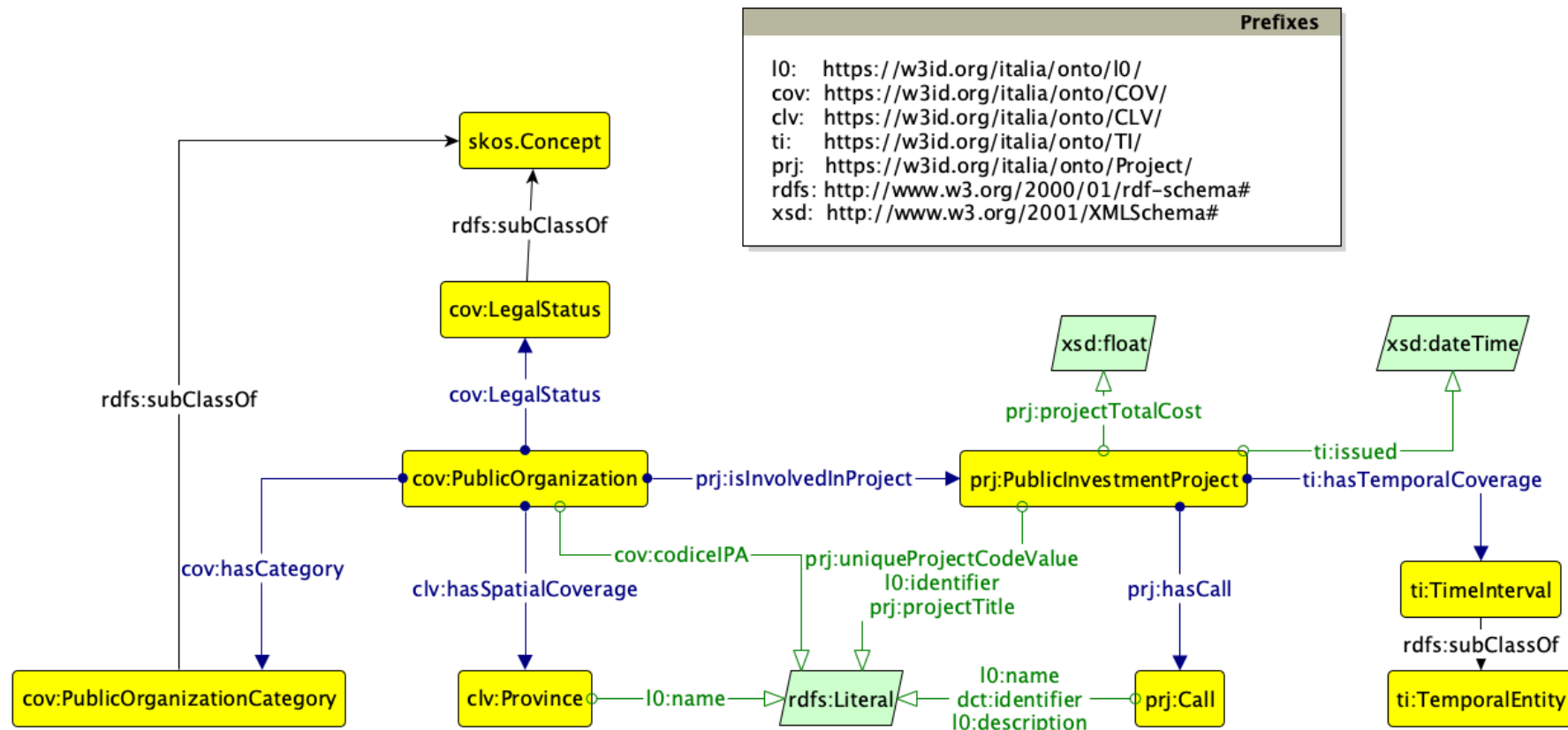
- Cosa sono i linked (open) data
  - Standard RDF
- Quali strumenti esistono per produrre linked (open) data
  - RML- RDF Mapping Language
- Il caso dei Linked Open Data dei finanziamenti del PNRR

Capire la semantica del dato tabellare

ORGANIZZAZIONE **PROGETTI**

	codice_ipa	comune	provincia	cod_provincia	regione	cod_regione	importo_finanziamento	avviso	data_invio_candidatura	data_finanziamento	codice_cup	numero_finestra_temporale	numero_di_protocoll
1.	istsc_siic80900g	Castelnuovo Berardenga	SI	052	Toscana	09	4977.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-06-01T11:33:40.000+0000	2022-09-05	G71C22000780006	2.0	31.0
2.	istsc_fgis001004	Monte Sant'Angelo	FG	071	Puglia	16	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-04-29T06:25:46.000+0000	2022-08-19	I71F22000290006	1.0	33.0
3.	istsc_toic852004	Montanaro	TO	001	Piemonte	01	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-04-29T10:42:42.000+0000	2022-08-19	C71F22000460006	1.0	33.0
4.	istsc_naic8cn00x	Torre del Greco	NA	063	Campania	15	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-05-05T08:40:08.000+0000	2022-08-19	D51F22000580007	1.0	33.0
5.	istsc_catd08000v	Cagliari	CA	092	Sardegna	20	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-06-07T06:19:21.000+0000	2022-08-19	B21F22001340006	2.0	33.0
6.	istsc_csic8ah00b	Corigliano-Rossano	CS	078	Calabria	18	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-04-28T17:46:36.000+0000	2022-08-19	D31F22000320006	1.0	33.0
7.	istsc_miic8cy00p	Milano	MI	015	Lombardia	03	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-05-17T10:53:59.000+0000	2022-08-19	G41F22000460006	1.0	33.0
8.	istsc_paps060003	Palermo	PA	082	Sicilia	19	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-06-20T08:27:44.000+0000	2022-08-19	E71F22001300006	2.0	33.0
9.	istsc_imic800005	Pieve di Teco	IM	008	Liguria	07	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-06-14T12:10:24.000+0000	2022-08-19	D71F22001430006	2.0	33.0
10.	istsc_tpic81800e	Campobello di Mazara	TP	081	Sicilia	19	12719.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-02T10:22:54.000+0000	2022-08-19	G11C22000250006	1.0	31.0
11.	istsc_ptic82000q	Monsummano Terme	PT	047	Toscana	09	3318.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-31T09:21:15.000+0000	2022-09-05	I91C22000670006	2.0	31.0
12.	icva_018	Pavia	PV	018	Lombardia	03	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-05-11T14:21:12.000+0000	2022-08-19	C11F22000330006	1.0	33.0
13.	istsc_aqmm02400x	Carsoli	AQ	066	Abruzzo	13	7742.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-19T08:14:53.000+0000	2022-08-19	H41C22000200006	1.0	31.0
14.	istsc_tptd03000e	Marsala	TP	081	Sicilia	19	12719.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-06-23T10:29:13.000+0000	2022-09-05	F81C22001170006	2.0	31.0
15.	istsc_rmic8fu00v	Roma	RM	058	Lazio	12	9954.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-10T14:30:38.000+0000	2022-08-19	G81C22000270006	1.0	31.0
16.	icgxx	San Ferdinando di Puglia	BT	110	Puglia	16	7301.0	1.4.1 Esperienza del Cittadino - Scuole - Aprile 2022	2022-04-29T07:45:26.000+0000	2022-08-19	B51F22000240006	1.0	33.0
17.	icvln	Vallo della Lucania	SA	065	Campania	15	8848.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-30T11:42:49.000+0000	2022-09-05	H71C22000900007	2.0	31.0
18.	istsc_pdic85700d	Lozzo Atestino	PD	028	Veneto	05	12719.0	1.2 Abilitazione al Cloud - Scuole - Aprile 2022	2022-05-09T08:00:38.000+0000	2022-08-19	E51C22000090006	1.0	31.0
19.	istsc_pdic85700d	Lozzo Atestino	PD	028	Veneto	05	7301.0	1.4.1 Esperienza del Cittadino - Scuole -	2022-05-02T18:30:38.000+0000	2022-08-19	E51F22000290006	1.0	33.0

Individuare ontologie di riferimento

- Ci sono due ontologie di schema.gov.it che possono essere utilizzate
  - Core Organization Vocabulary (COV) → **Organizzazioni**
  - Project (Progetti di ricerca e di investimento pubblico) → **Progetti**

Regole RML - organizzazioni - le sorgenti

- La prima cosa da fare è definire la/e sorgente/i dei dati da trasformare

```
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#>.
```

Prefissi

```
@base <https://w3id.org/italia/mapping/rml/orglocationFundedSchools> .

#############################################################
## LOGICAL SOURCES TO BE USED
#############################################################

##LogicalSourceFundedSchools
<#LogicalSourceFundedSchools> a rml:BaseSource ;
  rml:source "candidature_scuole_finanziate.csv";
  rml:referenceFormulation ql:CSV .
```

Definizione della BaseSource - il dataset CSV

Regole RML - la mappa di triple

- La seconda parte del file di regole è dedicata alla definizione della **mappa di triple**

  `soggetto – predicato – oggetto`

- La mappa di triple è costruita in base a delle regole che prendono i dati del CSV per rappresentarli in un grafo la cui semantica è data dalle ontologie di riferimento
  - Una mappa di triple deve sempre avere un soggetto (`rr:subjectMap`)

Regole RML - le organizzazioni

```
###########################################
# Organisation MAPPING
###########################################
<#OrganisationMapping> a rr:TriplesMap ;
  rml:logicalSource <#LogicalSourceFundedSchools> ;
  rr:subjectMap [
    rr:template "https://w3id.org/italia/data/public-organisation/{codice_ipa}" ;
    rr:class covapit:PublicOrganization , covapit:Organization , l0:Agent
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:IPACode;
    rr:objectMap [ rml:reference "codice_ipa" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate rdfs:label;
    rr:objectMap [ rr:template "La scuola {codice_ipa}" ; rr:language "it" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:hasLegalStatus;
    rr:objectMap [ rr:constant "https://w3id.org/italia/controlled-vocabulary/classifications-for-organizations/legal-status/26"
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:hasCategory;
    rr:objectMap [ rr:constant "https://w3id.org/italia/controlled-vocabulary/classifications-for-organizations/cofog-2009/09" ;
  ] .
```

Ma i dati geografici (province, comuni)?

- Esiste un vocabolario nazionale disponibile come linked open data sia per le province, sia per i comuni

- È possibile quindi fare linking tra questi vocabolari nazionali e il dataset linked open data da produrre

Ma i dati geografici (province, comuni)?

PROVINCE

- Per creare il collegamento con il vocabolario nazionale delle province si definisce un'altra sorgente nel file di regole
  - La sorgente è l'elenco delle province come risultano dal vocabolario attraverso un'apposita query SPARQL sull'endpoint di schema.gov.it

Mappare le province

```
##Logical Source RDF
<#InputSPARQL>
   a sd:Service ;
   sd:endpoint <https://schema.gov.it/sparql> ;
   sd:supportedLanguage sd:SPARQL11Query ;
   sd:resultFormat <http://www.w3.org/ns/formats/SPARQL_Results_JSON> .

### PROVINCE
<#LogicalSourcePROVINCERDF> a rml:BaseSource ;
    rml:source <#InputSPARQL>;
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.results.bindings[*]";
    rml:query """
    PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
    PREFIX clv: <https://w3id.org/italia/onto/CLV/>

    SELECT ?s ?provincia ?ID
    WHERE { ?s a skos:Concept ;
        skos:inScheme <https://w3id.org/italia/controlled-vocabulary/territorial-classifications/provinces> ;
        skos:notation ?ID ;
        clv:acronym ?provincia .

    } """
.
```

Sorgente SPARQL
Definisce l'endpoint
Il linguaggio supportato
E come resituire il risultato

La seconda sorgente dati è quindi lo sparql endpoint prima definito, come navigare i risultati restituiti in JSON (JSONPath) e la query

Mappare le province

```
<#OrganisationMapping> a rr:TriplesMap ;
  rml:logicalSource <#LogicalSourceFundedSchools> ;
  rr:subjectMap [
    rr:template "https://w3id.org/italia/data/public-org
    rr:class covapit:PublicOrganization , covapit:Organi
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:IPACode;
    rr:objectMap [ rml:reference "codice_ipa" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate rdfs:label;
    rr:objectMap [ rr:template "La scuola {codice_ipa}"
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:hasLegalStatus;
    rr:objectMap [ rr:constant "https://w3id.org/italia/
  ] ;
  rr:predicateObjectMap [
    rr:predicate covapit:hasCategory;
    rr:objectMap [ rr:constant "https://w3id.org/italia/
  ] ;
```

```
    rr:predicate clvapit:hasSpatialCoverage;
    rr:objectMap [
      rr:parentTriplesMap <#ProvinceMapping> ;
      rr:joinCondition [
        rr:child "cod_provincia";
        rr:parent "ID.value" ;
      ] ;
    ] ;
```

Parte di regole prima viste per le organizzazioni

Condizione di join tra due sorgenti di dati

Mappare le province

```
################################
## PROVINCE LINKING
################################
<#ProvinceMapping> a rr:TriplesMap ;
rml:logicalSource <#LogicalSourcePROVINCERDF> ;
rr:subjectMap [
  rr:template "https://w3id.org/italia/controlled-vocabulary/territorial-classifications/provinces/{ID.value}" ;  rr:termType rr:IRI ;
].
```

Il <#ProvinceMapping> è una mappa di triple con la sorgente data dalla query SPARQL prima definita e un soggetto che è l’URI del vocabolario controllato italiano costruito con il valore dell’ID restituito dalla query

Risultato in RDF Turtle

```
java -jar rmlmapper-6.0.0-r363-all.jar -m
MappingRMLOrgLocation.ttl -o ./scuole_pnrr.ttl -s turtle
```

```
<https://w3id.org/italia/data/public-organisation/21W263TE> a covapit:Organization,
    covapit:PublicOrganization, l0:Agent;
  rdfs:label "La scuola 21W263TE"@it;
  clvapit:hasSpatialCoverage <https://w3id.org/italia/controlled-vocabulary/territorial-classifications/provinces/081>;
  covapit:IPACode "21W263TE";
  covapit:hasCategory <https://w3id.org/italia/controlled-vocabulary/classifications-for-organizations/cofog-2009/09>;
  covapit:hasLegalStatus <https://w3id.org/italia/controlled-vocabulary/classifications-for-organizations/legal-status/26>;
```

RML Regole - organizzazioni coinvolte in progetti

```
##LogicalSourceFundedSchools
<#LogicalSourceFundedSchools> a rml:BaseSource ;
  rml:source "candidature_scuole_finanziate.csv";
  rml:referenceFormulation ql:CSV .
```

```
############################################
# Organisation linking project Mapping
############################################
<#OrganisationMapping> a rr:TriplesMap ;
  rml:logicalSource <#LogicalSourceFundedSchools> ;
  rr:subjectMap [
    rr:template "https://w3id.org/italia/data/public-organisation/{codice_ipa}" ;
  ] ;
  rr:predicateObjectMap [
    rr:predicate prjapit:isInvolvedInProject;
    rr:objectMap [ rr:template "https://w3id.org/italia/data/public-investment-project/{codice_cup}" ; rr:termType rr:IRI ]
  ] .
```

RML Regole - i progetti

```
<#ProjectMapping> a rr:TriplesMap ;
  rml:logicalSource <#LogicalSourceFundedSchools> ;
  rr:subjectMap [
    rr:template "https://w3id.org/italia/data/public-investment-project/{codice_cup}" ;
    rr:class prjapit:PublicInvestmentProject
  ] ;
  rr:predicateObjectMap [
    rr:predicate rdfs:label;
    rr:objectMap [ rr:template "Progetto di Investimento Pubblico nell'ambito del PNRR - codice CUP: {codice_cup}" ; rr:language "it" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate prjapit:uniqueProjectCodeValue;
    rr:objectMap [ rml:reference "codice_cup" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate prjapit:projectTotalCost;
    rr:objectMap [ rml:reference "importo_finanziamento" ; rr:datatype xsd:float ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate l0:identifier;
    rr:objectMap [ rml:reference "numero_di_protocollo" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate prjapit:projectTitle;
    rr:objectMap [ rr:template "Progetto {codice_cup} per la scuola {codice_ipa} relativo all'avviso {avviso}" ; rr:language "it" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate prjapit:hasCall;
    rr:objectMap <#NoticeURI>
  ] ;
  rr:predicateObjectMap [
    rr:predicate tiapit:issued;
    rr:objectMap [ rml:reference "data_invio_candidatura" ; rr:datatype xsd:dateTime ]
  ] ;
```

Collegare il progetto all’avviso PNRR

```
<#NoticeURI>
rr:termType rr:IRI ;
fnml:functionValue [
  rr:predicateObjectMap [
    rr:predicate fno:executes ;
    rr:objectMap [ rr:constant grel:array_join ]
  ];

   rr:predicateObjectMap [
     rr:predicate grel:p_array_a ;
     rr:objectMap [ rr:constant "https://w3id.org/italia/data/call"]
   ] ;

   rr:predicateObjectMap [
     rr:predicate grel:p_array_a ;
     rr:objectMap <#Md5ForNotice>
   ];
   rr:predicateObjectMap [
     rr:predicate grel:p_string_sep ;
     rr:objectMap [ rr:constant "/" ]
   ]
] .


<#Md5ForNotice>
rr:termType rr:IRI ;
fnml:functionValue [
   rr:predicateObjectMap [
     rr:predicate fno:executes ;
     rr:objectMap [ rr:constant grel:string_md5 ]
   ] ;
   rr:predicateObjectMap [
     rr:predicate grel:valueParameter ;
     rr:objectMap  [ rml:reference "avviso"]
   ]
  ].
```

L’URI dell’avviso è dato da il join di una costante (la prima parte dell’URI) con l’hash della stringa dell’avviso per l’ultima parte dell’URI

Uso della funzione MD5 per fare l’URI dell’avviso sulla base della stringa del CSV

Creare l’oggetto Avviso

```
<#CallMapping> a rr:TriplesMap ;
  rml:logicalSource <#LogicalSourceFundedSchools> ;
  rr:subjectMap [
    fnml:functionValue [
      rr:predicateObjectMap [
        rr:predicate fno:executes ;
        rr:objectMap [ rr:constant grel:array_join ]
      ];
       rr:predicateObjectMap [
         rr:predicate grel:p_array_a ;
         rr:objectMap [ rr:constant "https://w3id.org/italia/data/call"]
       ] ;
       rr:predicateObjectMap [
         rr:predicate grel:p_array_a ;
         rr:objectMap <#Md5ForNotice>
       ];
       rr:predicateObjectMap [
         rr:predicate grel:p_string_sep ;
         rr:objectMap [ rr:constant "/" ]
       ]
    ] ; rr:class prjapit:Call , l0:Object
  ] ;
  rr:predicateObjectMap [
    rr:predicate l0:description;
    rr:objectMap [ rml:reference "avviso" ; rr:language "it" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate l0:name;
    rr:objectMap [ rml:reference "avviso" ; rr:language "it" ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate l0:identifier;
    rr:objectMap <#substringMapping>
  ] .
```

La mappa di triple per l’oggetto di tipo Avviso
L’avviso ha una descrizione, un nome e un identificato
L’identificativo è ricavato prendendo la sottostringa della cella con la sola parte numerica

Risultato in RDF / Turtle

```
java -jar rmlmapper-6.0.0-r363-all.jar -m
MappingRMLOrgLocation.ttl RMLMappingProjectCall.ttl -o
./scuole_pnrr.ttl -s turtle
```

```
<https://w3id.org/italia/data/public-investment-project/B11C22000630006> a prjapit:PublicInvestmentProject;
  rdfs:label "Progetto di Investimento Pubblico nell'ambito del PNRR - codice CUP: B11C22000630006"@it;
  prjapit:hasCall <https://w3id.org/italia/data/call/a2b3eccd4c2083401f6c8c8d89bcf7f4>;
  prjapit:projectTitle "Progetto B11C22000630006 per la scuola istsc_lorc01000q relativo all'avviso 1.2 Abilitazione al Cloud - Scuole - A
  prjapit:projectTotalCost "2212.0"^^xsd:float;
  prjapit:uniqueProjectCodeValue "B11C22000630006";
  tiapit:hasTemporalCoverage <https://w3id.org/italia/data/time-interval/2022-09-05-2.0>;
  tiapit:issued "2022-06-23T10:54:35.000+0000"^^xsd:dateTime;
  l0:identifier "31.0" .
```

```
<https://w3id.org/italia/data/call/a2b3eccd4c2083401f6c8c8d89bcf7f4> a prjapit:Call,
      l0:Object;
  l0:description "1.2 Abilitazione al Cloud - Scuole - Aprile 2022"@it;
  l0:identifier "1.2";
  l0:name "1.2 Abilitazione al Cloud - Scuole - Aprile 2022"@it .
```

Alcuni elementi di attenzione nel caso PNRR

- Alcune colonne sono mappate su proprietà generiche perché non ci sono nelle ontologie delle specifiche proprietà che le possano rappresentare
  - E.g.: Numero di protocollo (che è comunque un identificativo)
- L'inserimento del codice ISTAT alfanumerico dei comuni ci consentirà di mapparli
  - Con solo le stringhe non è stato finora possibile
  - Il dataset Linked Open Data delle scuole sarà revisionato per la mappatura dei comuni

Grazie per l’attenzione!

giorgia.lodi@istc.cnr.it